DETAILED ACTION

1. This communication is responsive to Amendment filed 05/23/2008.

Claims 1-20 are pending in this application. Claims 1, 7, 14 are independent claims. In the Amendment, claims 19, 20 have been added, claims 1, 7, 14 have been amended.

EXAMINER'S AMENDMENT

2. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Applicant's representative Howard Levy, on 07/23/2008.

The application has been amended as follows:

- **Cancel claim 3**
- **Claim 1 has been amended as:**

A fixed length data search device, comprising:

a hash operation means for applying first and second similarly constructed hash function and thereby outputting multiple entry data corresponding to respective first and second hash values of an inputted fixed length datum;

a data table memory consisting of N numbers of memory banks, where N is an integer greater than or equal to two, the data table memory capable of storing a data table holding a large number of fixed length data;

a pointer table memory for storing a main memory pointer table, which is associated with the first hash function, and a subordinate memory pointer table for use when the main memory pointer table is filled to a predetermined level with respect to the N numbers of memory banks, which is associated with the second hash function that each indicates a memory address in said data table memory at which each fixed length datum is stored in said data table memory with said first and second hash values, each acting as a respective index therefore;

a pointer selector table to indicate which one of said main and subordinate memory pointer tables is referred to when a fixed length datum is inputted, wherein a datum identical to the single fixed length datum inputted to said hash operation means is searched in said data table through said hash operation means, said single fixed length datum registered in said data table if the datum has not been previously registered with said data table, and wherein if another fixed length datum having the same first hash value as an inputted fixed length datum has not been registered with said data table, said inputted fixed length datum is stored in said data table memory, and said memory address at which the datum is stored is managed with said main memory pointer table; and

a comparison means for simultaneously comparing a plurality of fixed length data stored at the same memory address in said N numbers of memory banks, the comparison means for outputting results of the comparison.

- **Claim 4 has been amended as:**

The fixed length data search device according to claim **1**, wherein each of a plurality of fixed length data having the same hash value are stored at the same memory address of a different memory bank in said data table memory.

- **Claim 5 has been amended as:**

The fixed length data search device according to claim **1**, wherein each of a plurality of fixed length data having a different hash value are stored at the same memory address of a different memory bank in said data table memory.

- **Cancel claims 8, 9, 12**
- **Claim 7 has been amended as:**

A fixed length data search device, comprising:

a hash operation means, said hash operation means using two types of similarly constructed hash functions to determine a first and second hash values of an inputted fixed length datum wherein said first and second hash values include multiple entry data;

a data table memory consisting of N numbers of memory banks, where N is an integer that is greater than or equal to two, the data table memory for storing a data table holding a large number of fixed length data;

a pointer table memory for storing a first memory pointer table, said pointer table memory that indicates a memory address in said data table memory

at which each fixed length datum is stored in said data table memory wherein said first hash value is an index, and a second memory pointer table for use when the first memory pointer table is determined to be filled to a predetermined level with respect to the N numbers of memory banks, which is associated with the second hash function, holding the memory address in said data table memory at which each fixed length datum is stored in said data table memory, said second hash value as an index;

a pointer selector table using said first hash value as an index to indicate which one of said first and second memory pointer tables is referred to when a fixed length datum is inputted, wherein when the number of stored data of separate fixed length data having the same first hash value exceeds N, a pointer in said pointer selector table corresponding to the first hash value of an unstored fixed length datum stored is set to said second memory pointer table, said memory address at which the datum is stored managed with said second memory pointer table, wherein if another fixed length datum having the same first hash value as an inputted fixed length datum has not been registered with said data table, said inputted fixed length datum is stored in said data table memory, and said memory address at which the datum is stored is managed with said main memory pointer table; and

a comparison means for simultaneously comparing a plurality of fixed length data stored at the same memory address in said N numbers of memory

banks, the comparison means for outputting results of the comparison.

- **Claim 10 has been amended as:**

The fixed length data search device according to claim **7**, wherein said comparison means comprises N numbers of comparators, said comparators simultaneously compare all bits to determine whether or not two fixed length data are identical.

- **Claim 11 has been amended as:**

The fixed length data search device according to claim **7**, wherein said comparison means determines if any of the fixed length data stored at the same memory address in said N numbers of memory banks matches the single fixed length datum inputted to said hash operation means and outputs the result of the determination.

- **Cancel claims 16, 19, 20**

Claim 14 has been amended as:

A method of searching fixed length data, comprising the steps of:

performing first and second similarly constructed hash operations to thereby outputting respective first and second hash values of inputted fixed length data, wherein each of said hash values includes multiple entry data;

referring to a main memory pointer table, which is associated with the first hash operation, or a subordinate memory pointer table for use when the main

memory pointer table is filled to a predetermined level with respect to N numbers of memory banks, which is associated with the second hash function, which is associated with the second hash operations, each of which holds a memory address in a data table memory at which each fixed length datum is stored in said data table memory with said first and second hash values each acting as a respective index therefor;

reading N numbers of fixed length data stored at an address pointed to by a pointer in said memory pointer table from a data table stored in said data table memory consisting of N numbers of memory banks, where N is an integer that is greater than or equal to two, the data table capable of storing a large number of fixed length data,

indicating in a pointer selector table which one of said main and subordinate memory pointer tables is referred to when a fixed length datum is inputted;

searching an identical datum to said inputted single fixed length datum in said data table based on its hash value, and registering said inputted single fixed length datum in said data table if said identical datum has not been detected in said step of searching;

detecting an exist bit associated with said data table into which the inputted single fixed length datum is to be registered;

determining if the exist bit indicates an on-state or an off-state of said data table,

when the exist bit indicates the on-state, proceeding with the registering,

when the exist bit indicates the off-state, proceeding with the second hash operation, and indicating that a next registered single fixed length datum is to be registered in the subordinate memory pointer table; and

simultaneously comparing said read N numbers of fixed length data with said inputted single fixed length datum, and outputting results of the comparison.

- **Claim 17 has been amended as:**

The method of searching fixed length data according to claim **14**, wherein each of separate fixed length data having the same hash value is registered with the same memory address of a different memory bank in said data table memory during said registering.

Reasons for Allowance

3. Claims 1-2, 4-7, 10-11, 13-15, 17-18 are allowed, now renumbered as 1-13.

4. The following is a statement of reasons for the indication of allowable subject matter:

The present invention relates to a device and method for searching fixed length data that have a fixed data bit length, and a computer program as well as a computer readable recording medium, and particularly to a device and method for rapidly searching a MAC address in an inter-network relay device, and a computer program as well as a computer readable recording medium. ([0001]).

Claim 1 recites, or similarly recites, in combination with the remaining elements, a computer system comprising:

a pointer table memory for storing a main memory pointer table, which is associated with the first hash function, and a subordinate memory pointer table for use when the main memory pointer table is filled to a predetermined level with respect to the N numbers of memory banks, which is associated with the second hash function that each indicates a memory address in said data table memory at which each fixed length datum is stored in said data table memory with said first and second hash values, each acting as a respective index therefore;

a pointer selector table to indicate which one of said main and subordinate memory pointer tables is referred to when a fixed length datum is inputted, wherein a datum identical to the single fixed length datum inputted to said hash operation means is searched in said data table through said hash operation means, said single fixed length datum registered in said data table if the datum has not been previously registered with said data table, and wherein if another fixed length datum having the same first hash value as an inputted fixed length datum has not been registered with said data table, said inputted fixed length datum is stored in said data table memory, and said memory address at which the datum is stored is managed with said main memory pointer table.

The closest prior art, et al. Gooch et al. (U.S. Pub No. 2003174710), shows a substantially similar method for performing flow based hash transformation to generate

hash pointers (Abstract). While Gooch teaches first and second similarly constructed hash values, Gooch does not explicitly teach the second hash function is not associated with subordinate memory pointer table for use when the main memory pointer table is filled to a predetermined level with respect to the N numbers of memory banks. Therefore, the cited arts, singularly or in combination, fail to anticipate or render the above cited limitations obvious.

Claim 7 recites, or similarly recites, in combination with the remaining elements, the method comprising the steps of:

a pointer table memory for storing a first memory pointer table, said pointer table memory that indicates a memory address in said data table memory at which each fixed length datum is stored in said data table memory wherein said first hash value is an index, and a second memory pointer table for use when the first memory pointer table is determined to be filled to a predetermined level with respect to the N numbers of memory banks, which is associated with the second hash function, holding the memory address in said data table memory at which each fixed length datum is stored in said data table memory, said second hash value as an index;

a pointer selector table using said first hash value as an index to indicate which one of said first and second memory pointer tables is referred to when a fixed length datum is inputted, wherein when the number of stored data of separate fixed length data having the same first hash value exceeds N, a pointer in said pointer selector table corresponding to the first hash value of an unstored fixed length datum stored is set to said second memory pointer table, said memory address at which the datum is stored managed with said second memory pointer table, wherein if another fixed length datum having the same first hash value as an inputted fixed length datum has not been registered

with said data table, said inputted fixed length datum is stored in said data table memory, and said memory address at which the datum is stored is managed with said main memory pointer table.

The closest prior art, et al. Gooch et al. (U.S. Pub No. 2003174710), shows a substantially similar method for performing flow based hash transformation to generate hash pointers (Abstract). While Gooch teaches first and second similarly constructed hash values, Gooch does not explicitly teach the second hash function is not associated with subordinate memory pointer table for use when the main memory pointer table is filled to a predetermined level with respect to the N numbers of memory banks. Therefore, the cited arts, singularly or in combination, fail to anticipate or render the above cited limitations obvious.

Claim 14 recites, or similarly recites, in combination with the remaining elements, the steps of:

referring to a main memory pointer table, which is associated with the first hash operation, or a subordinate memory pointer table for use when the main memory pointer table is filled to a predetermined level with respect to N numbers of memory banks, which is associated with the second hash function, which is associated with the second hash operations, each of which holds a memory address in said data table memory at which each fixed length datum is stored in said data table memory with said first and second hash values each acting as a respective index therefor;

reading N numbers of fixed length data stored at an address pointed to by a pointer in said memory pointer table from a data table stored in a data table memory consisting of N numbers of memory banks, where N is an integer that is greater than or equal to two, the data table capable of storing a large number of fixed length data,

indicating in a pointer selector table which one of said main and subordinate memory pointer tables is referred to when a fixed length datum is inputted;

searching an identical datum to said inputted single fixed length datum in said data table memory based on its hash value, and registering said inputted single fixed length datum in said data table memory if said identical datum has not been detected in said step of searching;

detecting an exist bit associated with said data table into which the inputted single fixed length datum is to be registered;

determining if the exist bit indicates an on-state or an off-state of said data table,

when the exist bit indicates the on-state, proceeding with the registering,

when the exist bit indicates the off-state, proceeding with the second hash operation, and indicating that a next registered single fixed length datum is to be registered in the subordinate memory pointer table.

The closest prior art, et al. Gooch et al. (U.S. Pub No. 2003174710), shows a substantially similar method for performing flow based hash transformation to generate hash pointers (Abstract). While Gooch teaches first and second similarly constructed hash values, Gooch does not explicitly teach the second hash function is not associated with subordinate memory pointer table for use when the main memory pointer table is filled to a predetermined level with respect to the N numbers of memory banks. Therefore, the cited arts, singularly or in combination, fail to anticipate or render the above cited limitations obvious.

5. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably

accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance".

Conclusion

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Miranda Le whose telephone number is (571) 272-4112. The examiner can normally be reached on Monday through Friday from 8:30 AM to 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John R. Cottingham, can be reached on (571) 272-7079. The fax number to this Art Unit is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Miranda Le/
Primary Examiner, Art Unit 2167